REMARKS/ARGUMENTS

After entry of this amendment, claims 1, 5-16, 18-19, and 24-49 will be pending in this application. Claim 1 has been amended to correct a typographical oversight. New claims 38-49 have been added. Support for the new and amended claims can be found in the specification. No new matter has been added.

Claims 1, 5-16, 18, 19, and 24-36 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Kim et al., Combined Data-Driven and Event-driven Scheduling Technique for Fast Distributed Cosimulation in view of Ghosh et al., A Hardware-Software Co-simulator for Embedded System Design and Debugging.

Reconsideration of these rejections and allowance of the pending claims in light of these remarks is respectfully requested.

Claim 1

Claim stands rejected under 35 U.S.C. 103(a) as being unpatentable over Kim in view of Ghosh. But this combination of references does not show or suggest each and every element of this claim. For example, claim 1 recites "receiving a variable synchronization parameter." The combination of references does not show or suggest this feature in combination with the other recited elements.

The pending office action cites Kim, section III, paragraph 4, as showing this feature. (See pending office action, page 6, fourth paragraph.)

Kim shows that one difficulty encountered when running cosimulations is that the software simulation can race ahead of the hardware simulation to the extent that a "past event" from the hardware simulation can reach the software simulation. (See Kim, section III, third paragraph.) These past events slow simulation times. (See Kim, section III, first paragraph.) Kim shows that there are two methods to solving this problem; the optimistic and conservative approaches. (*id.*)

In the optimistic approach, if a past event occurs, the software simulation rolls back to the time of the past event, then begins again. (See Kim, section III, second paragraph.)

Under the conservative approach, there are two schemes, the centralized and the distributed. (See Kim, section III, third paragraph.)

Kim shows that under the centralized approach, a "central controller manages the component simulators with the information on how far the local clock of the simulator can advance." (See Kim, section III, fourth paragraph.) How the central controller manages this information, or what the information includes, is not explored. (*id.*) Under the distributed approach, each simulator waits until it receives input events from all input ports. (See Kim, section III, fifth paragraph.) Kim further shows that a hybrid approach between these two is possible. (See Kim, section III, sixth paragraph.)

None of these approaches show or suggest <u>receiving a variable synchronization parameter</u>. Instead, Kim shows avoiding past events in one of three ways, by rolling back the software simulation, by managing information on how far the local clock can advance, or by waiting for inputs to be received at all inputs. None of these show receiving a variable synchronization parameter. Ghosh adds nothing to this. Accordingly, the combination of cited references does not show or suggest <u>receiving a variable synchronization parameter</u>, in combination with the other recited elements.

Claim 1 further recites "running the second software simulation system asynchronously with, and ahead of, the first software simulation system, wherein <u>the second software simulation system advances at most by a number of processor clock cycles set in the variable synchronization parameter</u> before the first software simulation advances by a clock cycle." The combination of references does not show or suggest this feature in combination with the other recited elements.

The pending office action cites Kim, section III, paragraph 4, as showing this feature. (See pending office action, page 6, fifth paragraph.)

Again, Kim shows that in one scheme that avoids simulation slowdowns due to past events, a "central controller manages the component simulators with the information on how far the local clock of the simulator can advance." (See Kim, section III, fourth paragraph.) There is no indication that the second software simulation system <u>advances at most by a number</u>

of processor clock cycles set in the variable synchronization parameter before the first software simulation advances by a clock cycle. Ghosh adds nothing to this. Accordingly, the combination of cited references does not show or suggest running the second software simulation system asynchronously with, and ahead of, the first software simulation system, wherein the second software simulation system advances at most by a number of processor clock cycles set in the variable synchronization parameter before the first software simulation advances by a clock cycle, in combination with the other recited elements.

Claim 1 further recites "the variable synchronization parameter limiting a maximum number of processor clock periods of the second simulation per period of a reference clock of the host machine." The combination of references does not show or suggest this feature in combination with the other recited elements.

The pending office action cites Kim, section III, paragraph 4, as showing this feature. (See pending office action, page 6, fifth paragraph.)

Again, Kim shows avoiding past events in one of three ways, by rolling back the software simulation, by managing information on how far the local clock can advance, or by waiting for inputs to be received at all inputs. None of these involve a variable synchronization parameter limiting a maximum number of processor clock periods of the second simulation per period of a reference clock of the host machine. Ghosh adds nothing to this. Accordingly, the combination of cited references does not show or suggest the variable synchronization parameter limiting a maximum number of processor clock periods of the second simulation per period of a reference clock of the host machine, in combination with the other recited elements.

Claim 1 further recites "controlling the first software simulation system using the second software simulation system." The combination of references does not show or suggest this feature in combination with the other recited elements.

The pending office action cites Kim, abstract, as showing this feature. (See pending office action, page 6, sixth paragraph.)

Kim shows a method of distributed cosimulation. (See Kim, abstract.) Kim further shows that this method uses intersimulator communications. (*id.*) However, Kim does

not show that this these intersimulator communications include one simulation system controlling another simulation system. Thus, Kim is silent that these intersimulator communications involve controlling the first software simulation system using the second software simulation system. Ghosh adds nothing to this. Accordingly, the combination of cited references does not show or suggest <u>controlling the first software simulation system using the second software simulation system</u>, in combination with the other recited elements.

For at least these reasons, claim 1 should be allowed.

<u>Other claims</u>

Claims 15 and 29 should be allowed for similar reasons as claim 1. The remaining rejected claims depend on one of the above claims and should be allowed for at least the same reasons and the additional limitations they recite.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 415-576-0200.

Respectfully submitted,

/ J. Matthew Zigmant /

J. Matthew Zigmant
Reg. No. 44,005

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 415-576-0200
Fax: 415-576-0300
JMZ:deb
61807527 v1